

EVALUATING MACHINE LEARNING REGRESSION MODELS FOR PREDICTING STOCK MARKET TRENDS FOR THE CONSTRUCTION INDUSTRY: A COMPARATIVE STUDY

N. Ashfaq¹, A. Khalid², N. Sarwar², I. Alam³, M. F. Afzal⁴ and J. Altaf⁵

¹Department of CS, University of Punjab

²Department of CS & IT, Bahria University, Lahore

³University of Sialkot, Sialkot, Pakistan

⁴Institute of Management Sciences Pak Aims Lahore, Pakistan

⁵Gulab Devi education complex Lahore, Lahore, Pakistan

ABSTRACT

The accurate anticipation of the stock market index's trajectory plays a pivotal role in formulating effective share trading methods. The accurate forecasting of closing stock prices has the potential to yield significant advantages for investors. Machine learning algorithms possess the capability to effectively analyze and predict historical stock patterns, hence yielding highly dependable closing price forecasts. This article presents a comprehensive study of the NASDAQ stock market, with a specific focus on constructing a diversified portfolio consisting of ten businesses from the construction industry as well as other industries. The aim of this study is to calculate the opening price of a stock of a construction enterprise or any other business for the following day by utilizing past data. In order to complete the assignment, we employed nine distinct Machine Learning regression models on a dataset consisting of stocks data. The performance of these models was assessed using Mean Squared Error (MSE) and the coefficient of determination (R^2).

Index Terms: Construction, Financial forecasting, Machine Learning, Regression Algorithms, Stock Market Prediction.

(Received 08.07.2024

Accepted 09.09.2024)

INTRODUCTION

The prediction of stock market performance poses a significant challenge for traders and analysts [1]. Individuals attempt to acquire shares at a lower cost and subsequently sell them at a higher value. The disparity between the sale price and the purchasing price is seen as a measure of profit. The business is regarded as proficient and portable by investors because to its high liquidity [2]. corporations engage in the sale of shares in order to raise funds for the purpose of financing new initiatives. Additionally, certain corporations devote a portion of their resources to invest in the stocks of other companies. The primary objective of this market is to generate financial gains. In order to enhance profitability, an investor endeavors to forecast forthcoming market trends [3]. Hence, financial forecasting has emerged as a prominent subject of interest in contemporary discourse. The majority of large firms are currently employing data scientists and financial analysts. While it is not possible to accurately forecast the precise figures, it is nonetheless feasible to make dependable predictions on the patterns [4], [5].

The prediction of trends in the stock market is heavily reliant on past time series data, as outlined in the

foundational principles of this domain [6]. A time series refers to a sequence of numerical data points arranged in chronological order. The selected data points are monitored for their movements at regular time intervals [7]. The data pertaining to the stock market is generated on a daily basis. Time series analysis involves the utilization of several statistical approaches to construct models that capture the inherent properties of time series data [8]. This procedure typically entails making assumptions about the characteristics of the data and subsequently breaking it down into smaller constituent parts. This study holds significance as it enables the assessment of past time series data for the purpose of forecasting future trends [9]. By utilizing past trends, investors are able to enhance their decision-making capabilities.

This study employs nine distinct machine learning models to conduct time series analysis on stock market data. The present study employs a range of regression models, including Support Vector Regressor(SVR) [36], Ridge Regression(RR) [38], Linear Regression(LR) [37], ElasticNet(EN) [40], Lasso Regression(Lasso) [39], Decision Tree Regressor(DTR) [41], Extra Tree Regressor(ETR) [43], Random Forest Regressor(RFR) [42], and RANSAC [44]. The objective

is to determine the most effective model for analyzing financial time series data. The objective of this study is to forecast the opening price of stock data for the following day, specifically focusing on five distinct sectors that encompass a total of ten stocks.

The subsequent allocation of the remaining paper is as follows. Section two encompasses a comprehensive review of prior research conducted on the analysis of time series data pertaining to equities. Section three provides an overview of the process of preparing the dataset, while section four delves into the methodology employed in the study. The evaluation of models is conducted in section five, while the discussion on the outcomes is included as part of section six. The paper is concluded in section seven, which also includes a discussion of the limits and suggestions for further research.

LITERATURE REVIEW

The act of forecasting financial market trends contradicts the fundamental principles of finance, as outlined by the Efficient Market Hypothesis (EMH). According to this concept, the utilization of historical data for predicting future trends may yield no profits if the entire market becomes aware of that pattern, leading to a self-correcting market phenomenon [10]. Numerous academics have already refuted this concept through the utilization of diverse indicators and algorithms.

Predicting stock prices is a critical goal in the finance field [11], [12]. In 2015, Shetaet et al. conducted a study comparing the performance of artificial neural network (ANN) and SVR algorithms on time series data of the S&P 500 index. The results showed that SVR outperformed ANN [13]. In 2015, Roy et al. conducted a study on Goldman Sachs Group Inc.'s stocks' time series data. They used LR and found that their suggested model performed better than RR [14]. Aseervatham et al. asserted that RR can attain equivalent performance to the SVR [15]. Nair et al. introduced a hybrid automated system for stock market prediction that combines DT and adaptive neuro-fuzzy techniques [16]. Gunter and Senyurt utilised Random Forest and CART (Classification and Regression Tree) algorithms to analyse a decade's worth of technical indicator data [17]. Thakur et al. created a stock price recommendation system by employing various regression models, including Lasso linear regression, RR, LR, SVR, KNN, and RF. Among these models, SVR has shown superior performance [18]. Nunno et al. conducted research on Apples.inc's stocks by citeb19. Wu et al. developed a recommendation system for determining the optimal timing to buy or not buy a stock. They achieved this by implementing a two-layer bias decision tree that incorporates technical indicator features [20]. Huang et al. introduced a model that combines the filter rule and

DT technique for predicting the stock market [21]. Yang et al. conducted a study on the asymptotic properties of the adaptive elastic net in ultrahigh-dimensional sparse linear regression models. They also introduced a new model with the aim of enhancing prediction accuracy [22].

Gharehchopogh et al. introduced a LR model to forecast stock market trends. They demonstrated that this model performs similarly to the actual trading volume, as evidenced by the R^2 performance measure [23]. Sathyaraj, R., conducted a comparison analysis of three regression algorithms (LR, RR, and SVM) on nine distinct stocks to forecast the price for the following day. The models' accuracy has been assessed using R^2 and RMSE metrics [24]. Kazem et al. chose three time series data sets of stocks from NASDAQ in order to assess the accuracy of the proposed model. They have developed a novel forecasting model that uses SVR, chaotic mapping, and the firefly algorithm to predict stock market prices. The performance measurements used are MSE and Mean Absolute Percent Error (MAPE) [25]. Guresen et al. conducted a comparative analysis to estimate the daily rate value of the NASDAQ stock index [26]. Weng et al. used a variety of data sources to forecast the movement of NASDAQ Apple stocks one day in advance [27]. Di et al. have selected three datasets: Apple Inc., Microsoft Corporation, and Amazon.com, Inc. He utilised the very randomised tree technique for feature selection. The extracted features are subsequently utilised to train a SVM model, which is employed to forecast the future trends for the next 3, 5, 7, and 10 days [28].

Smith investigated the application of linear regression algorithms to forecast Apple Inc.'s stock values. The study highlighted the efficacy of linear regression in calibrating weights for different features to predict forthcoming stock values. This study emphasised the capability of basic regression models to generate accurate predictions when properly calibrated [45]. S,ekeroglu et al. compared different machine learning and deep learning algorithms, like Random Forest Regression, to see how well they could predict the stock prices of companies with dual-listed stocks. The study revealed that deep learning models exhibited great accuracy, but machine learning models such as Random Forest were similarly proficient, especially when confronted with a substantial number of features [46]. Easton et al. utilised the k-NN algorithm to predict earnings for the next one, two, and three years [47]. Bloch performed an extensive examination of various machine-learning techniques for forecasting stock prices, with particular emphasis on regression models. The study examined multiple algorithms, such as linear regression, and determined that no individual method demonstrated superior performance in all situations. Conversely, the performance was contingent upon the distinct attributes of the dataset and the stock under examination [48].

METHODOLOGY

This study conducted a systematic evaluation of the predictive performance of nine distinct machine learning regression models for stock price prediction.

Figure 1 shows the whole process of stock market prediction. Starting with data Scrapping to preprocessing, Feature selection, input to regression algorithms, tesing, performance measure of the models and comparisons on the basis of results of these performance measure.

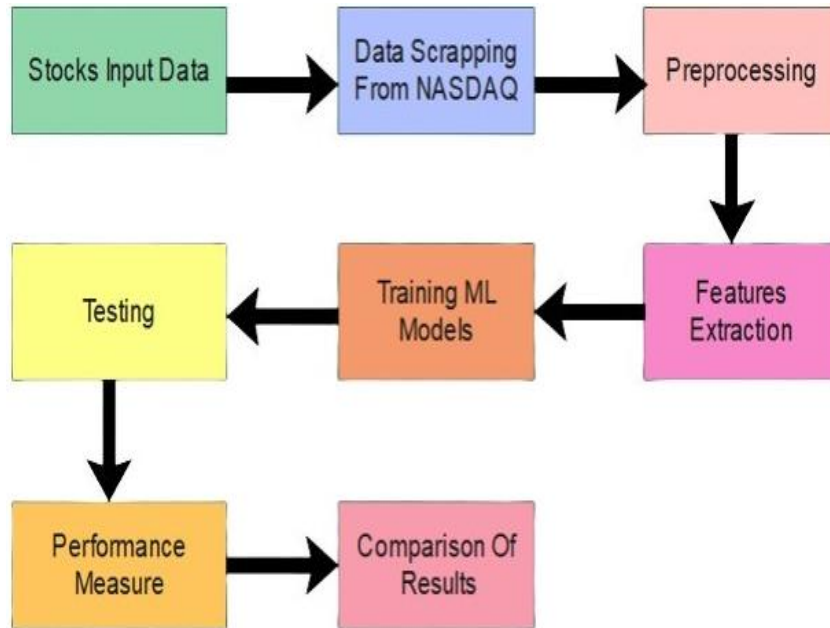


Fig. 1: Workflow for Stock Market Prediction Using Machine Learning.

A. Datasets: The target market for this comparison is the NASDAQ, which includes companies from around the world, not just those headquartered in the U.S. [29] The researcher gathered historical data, including daily close, open, low, and high prices, from the NASDAQ website. The Python library BeautifulSoup4 was employed for web scraping. The objective of this research is to develop a model that can accurately predict the opening price for the next trading day.

$target = current\ day\ closing\ price - next\ day\ opening\ price$

In order to demonstrate the diversity of this study, a selection of ten stocks from five distinct sectors was made. These stocks were chosen randomly and have a historical data spanning a period of ten years. The details of target companies are shown in Table I.

Table I: List of companies, sectors, industries, and their respective stock symbols analyzed in this study.

Sector	Industry	Company	Symbol
Finance	Major Bank	Ameris Bancorp	ABCB
	Property Casualty Insurer	Amerisafe, Inc.	AMSF
Technology	Computer Software	Microsoft Corporation	MSFT
	Computer Manufacturing	Apple Inc.	AAPL
Energy	Coal Mining	Alliance Holdings GP, LP	AHGP
Industrial	Machinery/Components	American Electric Technologies, Inc.	AETI
Health Care	Major Pharmaceuticals	ACADIA Pharmaceuticals Inc.	ACAD
	Medical/Dental Instruments	Accuracy Incorporated	ARAY
Consumer	Other Specialty Store	1-800Flowers.com, Inc.	FLWS
Service	Catalog/Specialty Distribution	Amazon.com, Inc.	AMZN

Each stock has a data size of 2519 sample points. 80% of the stock is allocated for training, while the remaining portion is reserved for testing.

B. Support Vector Regression(SVR): SVR is a type of supervised machine learning method that can be utilised for tasks such as classification, regression, or outlier detection. The process involves the creation of a hyperplane or a collection of hyperplanes in a space with a large number of dimensions. The model preserves all the key attributes that define the algorithm. The selection of training data is only determined by a subset due to the cost function used in model development, which disregards any training data that closely aligns with the model's prediction [30]. The objective of SVR is to find a function $f(x) = \langle w, x \rangle + b$ that minimizes the following cost function:

$$\min_{w, b, \xi, \xi^*} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n (\xi_i + \xi_i^*)$$

Subject to:

$$y_i - \langle w, x_i \rangle - b \leq \epsilon + \xi_i, \quad \langle w, x_i \rangle + b - y_i \leq \epsilon + \xi_i^*$$

Where, w is the weight vector, b is the bias term, ξ_i and ξ_i^* are slack variables representing deviations from the margin. C is the regularization parameter, controlling the trade-off between margin maximization and error minimization. ϵ defines the width of the margin around the hyperplane within which no penalty is given to errors.

C. Linear Regression(LR)

LR involves estimating the value of a dependent variable y based on an independent variable X , assuming that there is a linear connection between the two variables. The relationship can be graphically depicted as a linear function. When there are numerous independent variables, the procedure is referred to as multiple linear regression [31]. The Linear Regression model can be expressed as:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_p x_p + \epsilon$$

Where, y is the dependent variable (target), $x_1, x_2, \dots, x_p$ are the independent variables (features), β_0 is the intercept and $\beta_1, \beta_2, \dots, \beta_p$ are the coefficients for the respective features. ϵ is the error term (residual), which accounts for the variability in y that cannot be explained by the linear relationship.

The coefficients $\beta_0, \beta_1, \dots, \beta_p$ are typically estimated using the Ordinary Least Squares (OLS) method, which minimizes the sum of squared residuals:

$$\min_{\beta} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

Where, y_i is the observed value of the dependent variable and $\hat{y}_i$ is the predicted value of the dependent variable based on the model.

D. Lasso Regression

Lasso is a variation of the Least Square Method that performs exceptionally well when the number of characteristics is less than the number of observations.

The process involves generating solutions by estimating coefficients that are sparsely distributed. The L1 norm is employed, which is equal to the absolute value of the magnitude of the coefficients. It performs feature selection and shrinkage by reducing the coefficients of other features to zero. [14]. The model can be formulated as:

$$\min_{\beta} \left\{ \frac{1}{2n} \sum_{i=1}^n \left(y_i - \left(\beta_0 + \sum_{j=1}^p \beta_j x_{ij} \right) \right)^2 + \lambda \sum_{j=1}^p |\beta_j| \right\}$$

Where, y_i is the dependent variable (target), x_{ij} are the independent variables (features), β_0 is the intercept, β_j are the coefficients for the respective features and λ is the regularization parameter that controls the strength of the penalty.

The objective function combines the least squares error with a penalty term $\lambda \sum_{j=1}^p |\beta_j|$, which is the L_1 norm of the coefficient vector. As the value of the regularisation parameter λ grows, a greater number of coefficients are pushed towards zero. This might result in the creation of simpler models that have a reduced risk of overfitting, but at the cost of introducing bias.

$$\hat{\beta} = \arg \min_{\beta} \left\{ \frac{1}{2n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^p |\beta_j| \right\}$$

The given formulation enables Lasso to achieve a trade-off between accurately fitting the data and maintaining tiny or zero model coefficients. This capability aids in addressing multicollinearity issues and enhancing the interpretability of the model.

E. Ridge Regression(RR)

RR is a type of regularised linear regression that is particularly effective when the number of features is smaller than the number of data. This tool belongs to the class of regression techniques that utilise L_2 regularisation, which involves adding an L_2 penalty equal to the square of the magnitude of the coefficients. It is unable to nullify coefficients, therefore it either includes all coefficients or excludes them entirely [14]. The model can be formulated as:

$$\min_{\beta} \left\{ \frac{1}{2n} \sum_{i=1}^n \left(y_i - \left(\beta_0 + \sum_{j=1}^p \beta_j x_{ij} \right) \right)^2 + \lambda \sum_{j=1}^p \beta_j^2 \right\}$$

Where, y_i is the dependent variable (target), x_{ij} are the independent variables (features), β_0 is the intercept, β_j are the coefficients for the respective features and λ is the regularization parameter that controls the strength of the penalty. As the value of the regularisation parameter λ grows, the coefficients are compressed towards zero, hence mitigating the risk of overfitting. Unlike Lasso Regression, Ridge Regression does not engage in variable selection as it does not impose a need for any coefficients to reach a value of zero.

$$\hat{\beta} = \arg \min_{\beta} \left\{ \frac{1}{2n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^p \beta_j^2 \right\}$$

The aforementioned formulation of Ridge Regression offers a means to achieve a balance between accurately fitting the training data and preserving model simplicity. This balance ultimately results in improved generalisation performance when applied to unknown data.

F. Elastic Net Regression(ER)

The EN model combines characteristics from both Ridge and LASSO Regression models. It performs effectively on a larger dataset. The model utilises multicollinearity and regularisation values in its optimisation function, resulting in an inherent rise in overfitting and an improvement in the predictive capacity of the model. The panelization of the system is achieved by using both L_1 (Lasso) and L_2 (Ridge), resulting in the effective reduction of certain coefficients and the setting of others to zero [32]. The model can be formulated as:

$$\min_{\beta} \left\{ \frac{1}{2n} \sum_{i=1}^n \left(y_i - \left(\beta_0 + \sum_{j=1}^p \beta_j x_{ij} \right) \right)^2 + \lambda \left(\alpha \sum_{j=1}^p |\beta_j| \right) \right\}$$

Where, y_i is the dependent variable (target), x_{ij} are the independent variables (features), β_0 is the intercept, β_j are the coefficients for the respective features, λ is the regularization parameter that controls the overall strength of the penalty. α is the mixing parameter that controls the balance between L_1 and L_2 regularization. When $\alpha = 1$, Elastic Net is equivalent to Lasso Regression. When $\alpha = 0$, Elastic Net is equivalent to Ridge Regression. For values between 0 and 1, Elastic Net is a combination of both Lasso and Ridge.

$$\hat{\beta} = \arg \min_{\beta} \left\{ \frac{1}{2n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 + \lambda \left(\alpha \sum_{j=1}^p |\beta_j| + \frac{1-\alpha}{2} \sum_{j=1}^p \beta_j^2 \right) \right\}$$

The formulation mentioned above enables Elastic Net to take advantage of the benefits of both Lasso and Ridge, making it a reliable option for regression tasks in intricate datasets.

G. RANSAC

RANSAC is an iterative technique that is non-deterministic and is used to estimate parameters from a subset of inliers within a dataset that may contain outliers. The data is partitioned into two distinct categories: outliers and inliers. The inliers are subsequently utilised for the model [35]. Let $\mathbf{x}_i$ be a data point and $M(\theta)$ be a model parameterized by θ . The RANSAC algorithm seeks to solve the following:

- Randomly select a small subset of data points $S \subset \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$.
- Estimate model parameters θ using only the points in S .
- Define the consensus set as:

$$C = \{\mathbf{x}_i | \text{error}(\mathbf{x}_i, M(\theta)) < \epsilon\}$$

where ϵ is a predefined threshold for error.

- If the size of the consensus set $|C|$ is greater than a threshold T , re-estimate θ using all the points in C .
- Iterate the process and select the model parameters θ with the largest consensus set as the final model.

H. Decision Tree (DT)

The DTR algorithm constructs a tree iteratively by partitioning the dataset into subsets, resulting in a tree composed of decision nodes and leaf nodes. A decision node contains multiple branches, each reflecting different attribute values that are being checked. A leaf node represents the final conclusion made regarding the numerical target. The highest node in the hierarchy is referred to as the root node, which correlates to the most accurate predictor [44].

In order to determine the division of data at each node, a commonly employed criterion is MSE. A DT is created by dividing the data into smaller subsets through a recursive process. At every node, the algorithm assesses all potential divisions based on the attributes and chooses the one that minimises the MSE. The approach computes the following for each feature X_j and split point s in a formal manner:

$$\text{Split Cost} = \frac{n_{\text{left}}}{n} \cdot \text{MSE}_{\text{left}} + \frac{n_{\text{right}}}{n} \cdot \text{MSE}_{\text{right}}$$

Where, n_{left} and n_{right} are the number of samples in the left and right child nodes, respectively. MSE_{left} and $\text{MSE}_{\text{right}}$ are the mean squared errors in the left and right child nodes, respectively. The algorithm selects the feature and split point that yield the minimum split cost. This procedure is then continued recursively until a stopping requirement is satisfied, such as reaching a maximum depth or having too few samples in the node.

$$\hat{y} = \frac{1}{n_{\text{leaf}}} \sum_{i=1}^{n_{\text{leaf}}} y_i$$

Where n_{leaf} is the number of observations in the leaf node.

I. Extra Tree(ET)

The ETR, short for exceptionally randomised trees, differs from standard decision trees due to its unique node splitting technique. The algorithm randomly performs splits for each of the maximum features and selects features randomly. It then chooses the best split among those. Setting max-feature to 1 will result in the construction of a completely random decision tree on each iteration [34].

The goal at each node is to divide the data in a way that minimises the MSE, albeit the splits are selected in a more random manner compared to traditional decision trees. The split cost is computed by taking into account a feature X_j and a random threshold s .

$$\text{Split Cost} = \frac{n_{\text{left}}}{n} \cdot \text{MSE}_{\text{left}} + \frac{n_{\text{right}}}{n} \cdot \text{MSE}_{\text{right}}$$

Where, n_{left} and n_{right} are the number of samples in the left and right child nodes, respectively. MSE_{left} and $\text{MSE}_{\text{right}}$

are the mean squared errors in the left and right child nodes, respectively.

J. Random Forest(RF)

RF is a collection of many binary regression trees. A group of variables that are not dependant on each other is utilised for the development of a substantial number of binary regression trees. The decision trees are constructed using bootstrapped samples from the dataset, while Random Forest randomly chooses variables to split. [6]. Let T be the total number of trees in the Random Forest. Each tree t produces a prediction $\hat{y}^{(t)}$ for a given input $\mathbf{x}$. The final prediction $\hat{y}$ is the average of the predictions from all the trees:

$$\hat{y} = \frac{1}{T} \sum_{t=1}^T \hat{y}^{(t)}$$

At every node, the optimal division is determined by evaluating a random subset of characteristics. The introduction of randomization aids in decorrelating the trees, hence diminishing the model's variance.

K. Experimental Setup

The experimental configuration utilised Python 3.9 on Ubuntu 22.04, incorporating the following libraries: BeautifulSoup4 4.12.2, NumPy 1.25.0, Pandas 2.0.3, urllib, requests 2.31.0, parse, and re. Nine distinct supervised machine learning regressors were utilised to train the models and ascertain the polarity of the test dataset. The dataset was trained and predicted using Scikit-learn version 1.3.0. This program provides the capacity to be reused in many situations and is easy for users to navigate, with a wide range of machine learning regressors available.

L. Performance Measures

There are various evaluation metrics available for assessing the predictions of Machine Learning regression algorithms. In our study, we used MSE and the coefficient of determination R^2 to evaluate the performance of our models.

1) **Mean Square Error:** MSE quantifies the size of an error by computing the average of the squared discrepancies between the desired values and the estimated values. It is often recognized as a risk function that evaluates the accuracy of an algorithm's predictions. The MSE number is always non-negative, and a lower value indicates greater performance of the algorithm.

$$MSE(target, predicted) = \frac{1}{N} \sum (target - predicted)^2$$

Where N is the total numbers of sample points.

2) **R^2 Error :** The R^2 value, or coefficient of determination, quantifies the predictive accuracy of an algorithm for future outcomes. If the worst-case scenario occurs, the model may exhibit poor performance. A value of R^2 ranging from 0 to 1 reflects the degree of accuracy

in the fit, where 0 indicates no fit and 1 indicates a flawless fit. As the value increases, the algorithm's prediction performance improves.

$$R^2(target, predicted) = 1 - \frac{\sum (target - predicted)^2}{\sum (target - targetMean)^2}$$

RESULTS AND DISCUSSIONS

We performed a comparison investigation of 9 separate regression algorithms using datasets from 10 organizations, each belonging to a different economic sector. The aim was to determine the most effective regression algorithm for these specific datasets, offering useful insights for academics and a clear guidance for newcomers. Table II presents a summary of the results. Each column in the table represents a single stock, and each row provides information on the Mean Squared Error (MSE) and R^2 values for the corresponding algorithms. Figures 2 and 3 depict the MSE and the R^2 error plots, respectively.

When these algorithms were utilised on the data of MSFT, a technology firm, the DT and ET models yielded the most favourable outcomes on the training data, attaining a MSE of 0.0000 and a R^2 of 1.0000. On the other hand, Lasso and RANSAC had the poorest performance, as indicated by their MSE values of 0.1175 and 0.1344, and R^2 values of 0.0000 and -0.1444, respectively. SVM and Lasso yielded the most favourable outcomes on the test data, exhibiting MSE values of 0.2642 and 0.2639, and R^2 values of -0.0078 and -0.0066, correspondingly. In contrast, RANSAC and LR demonstrated the lowest performance on the test data, achieving MSE values of 0.4951 and 0.4672, and R^2 values of -0.0291 and -0.8887, respectively. Based on the data from AAPL, a technological business, the DT and ET algorithms yielded the most favourable outcomes on the training data. They achieved a MSE of 0.0000 and a R^2 of 1.0000. On the other hand, Lasso and RANSAC exhibited the poorest performance, as indicated by their MSE values of 0.6624 and 0.6769, and R^2 values of 0.0002 and -0.0217, respectively. Similarly, when evaluated on the test data, EN and SVM exhibited the most favourable outcomes, achieving MSE values of 1.1796 and 1.1801, and R^2 values of 0.1582 and -0.0017, respectively. In contrast, LR and DT had the lowest performance on the test data, with MSE values of 2.5634 and 1.6138, and R^2 values of -0.0294 and -0.3698, respectively. During the training of the algorithms using AMSF data, DT and ET consistently showed the highest performance on the training data.

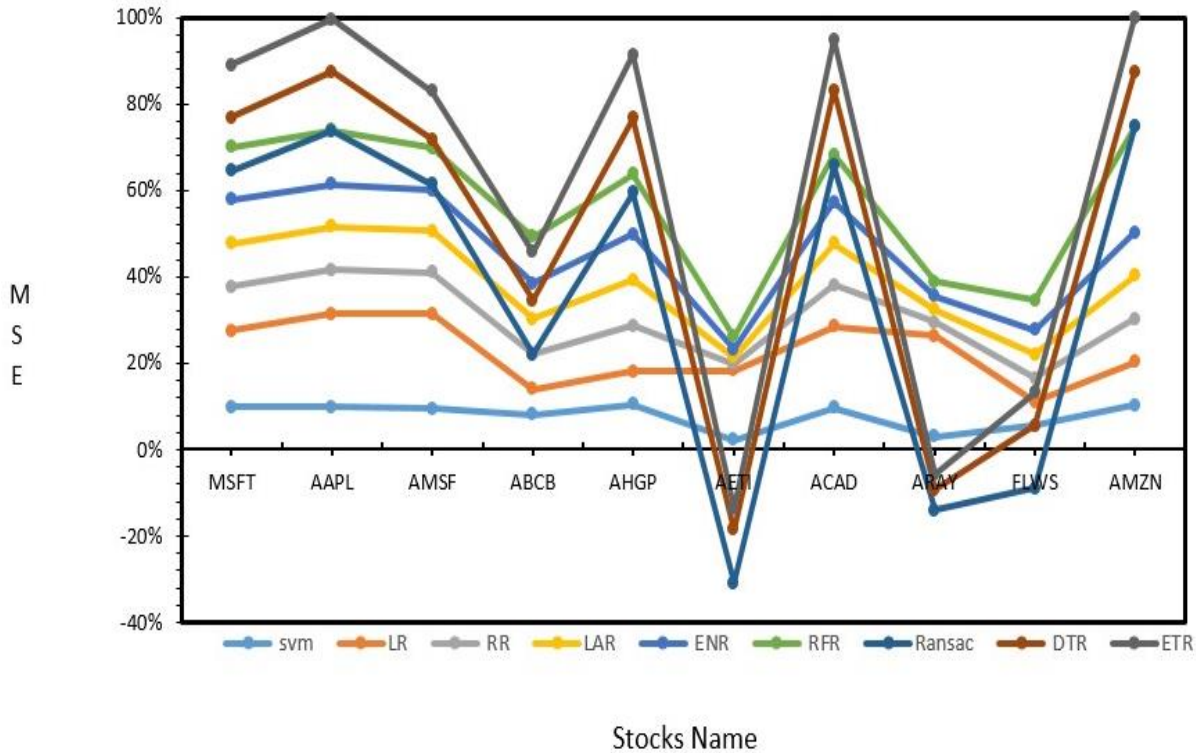


Fig. 2: Mean Square Error of all algorithms for stocks data.

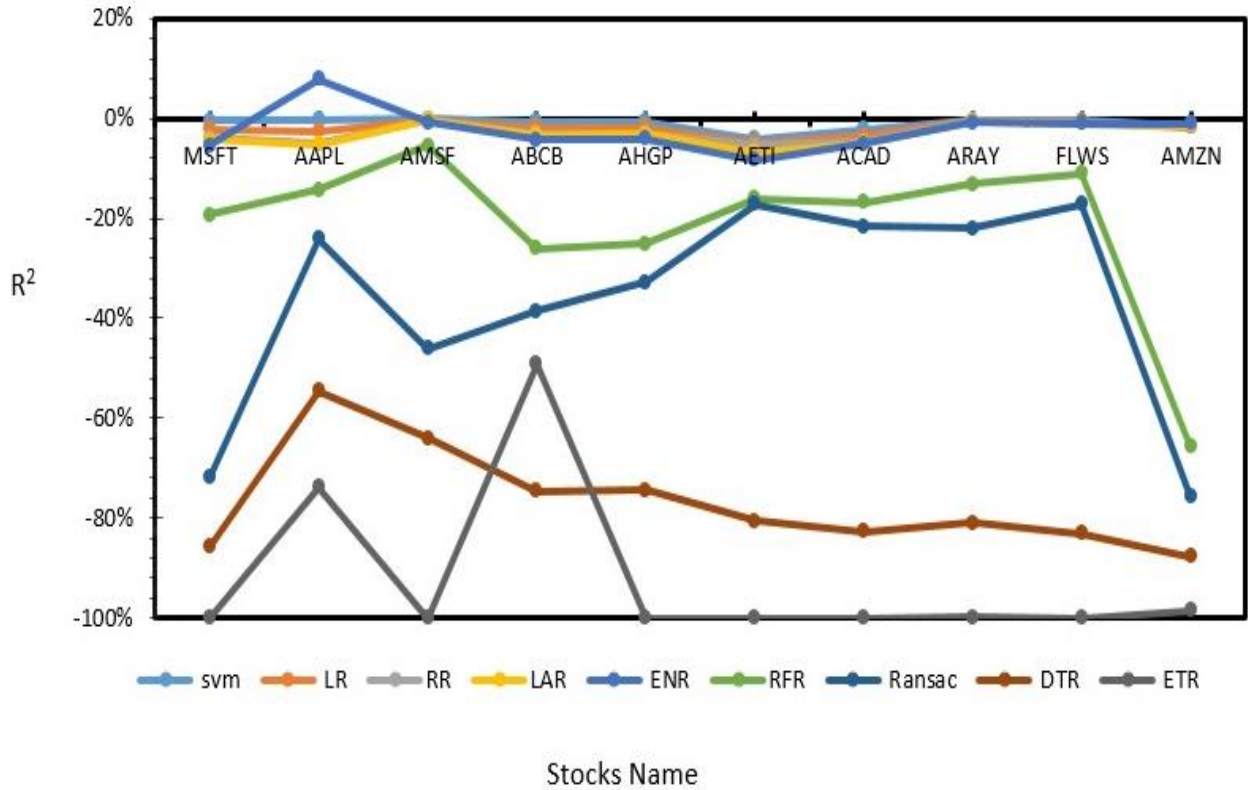


Fig. 3: R² Error of all algorithms for stocks data.

They achieved MSE values of 0.0000 and R^2 values of 1.0000. LR and RANSAC had the poorest performance, with MSE values of 0.4339 and 0.1133, and R^2 values of 0.0057 and -0.1528, respectively. Among the test data, SVM and Lasso had the most superior performance, achieving a MSE of 0.1708 and a R^2 of -0.0000. However, RANSAC and LR had the lowest performance, as indicated by their MSE values of 0.2032 and 0.4339, and R^2 values of -0.1901 and -0.0007, respectively.

Out of all the algorithms used on ABCB data, the DT and ET models had the highest performance on the training data. They achieved a MSE of 0.0000 and a R^2 of 1.0000. Linear Regression and RANSAC had the poorest performance, with MSE values of 0.0499 and 0.0368, and R^2 values of 0.0059 and -0.4673, respectively. LR and Lasso yielded the most favourable outcomes on the test data, exhibiting MSE values of 0.0996 and 0.1923, and R^2 values of -0.0181 and -0.0056, respectively. In contrast, the DT and ET models showed the lowest performance on the test data, with MSE values of 0.2146 and 0.1923, and R^2 values of -0.5436 and -0.3826, respectively. After implementing these algorithms on the data of AHGP, a coal mining firm, the DT and ET models yielded the most favourable outcomes on the training data. They achieved a mean squared error MSE of 0.0000 and a R^2 of 1.0000. LR and RANSAC had the poorest performance, with Mean Squared Error MSE values of 0.00481 and 0.0265, and R^2 values of 0.0059 and -0.0517, respectively. LR and Lasso achieved the most favourable outcomes on the test data, with MSE values of 0.0996 and 0.1398, and R^2 values of -0.0181 and -0.0056, respectively. In contrast, the DT and ET models showed the lowest performance on the test data, with MSE values of 0.2300 and 0.1946, and R^2 values of -0.6543 and -0.3994, respectively. The DT and ET models had the highest performance on the training data of AEIT, an industrial machinery company, with a MSE of 0.0005 and a coefficient of determination R^2 of 0.9681. LR and RANSAC had the lowest performance, with MSE values of 0.00487 and 0.0179, and R^2 values of 0.0470 and -0.2006, respectively. RR and Lasso yielded the most favourable outcomes on the test data, exhibiting mean MSE values of 0.0057 and 0.0059, and R^2 values of -0.0822 and -0.1357, correspondingly. In contrast, the DT and LR models showed the lowest performance on the test data, as indicated by their MSE values of 0.0433 and 0.0567, and R^2 values of -7.2923 and -0.0804, respectively.

After being trained on ACAD data, the DT and ET models demonstrated the highest performance on the training data. They achieved a MSE of 0.0000 and a R^2 of 1.0000. On the other hand, Lasso Regression and RANSAC had the poorest performance, as indicated by

their MSE values of 0.1478 and 0.1594, and R^2 values of 0.0000 and -0.0786, respectively. Similarly, when applied to the test data, EN and Lasso yielded the most favourable outcomes, achieving MSE values of 0.2904 and 0.2910, and R^2 values of -0.0013 and -0.0034, respectively. In contrast, the DT and LR models showed the lowest performance on the test data, with MSE values of 0.5316 and 0.5728, and R^2 values of -0.8331 and -0.0170, respectively.

When evaluating the algorithms for ARAY, a firm specialising in medical and dental instruments, the DT and ET models demonstrated superior performance on the training data. These models achieved a MSE of 0.0000 and a R^2 of 1.0000. LR and RANSAC exhibited subpar performance, as indicated by their respective MSE values of 0.0897 and 0.0126, and R^2 values of 0.0038 and -0.1031. During the test, EN, Lasso, and Ridge showed better performance compared to other methods. They achieved MSE values of 0.0059 and R^2 values of -0.0019, -0.0016, and 0.0002, respectively. In contrast, the DT and LR models showed lower performance on the test data, as indicated by their MSE values of 0.0097 and 0.0455, and R^2 values of -0.5301 and 0.0007, respectively.

The DT and ET models achieved the best results on the training data of FLWS, a speciality store. They achieved a MSE of 0.0000 and a R^2 of 1.0000. On the other hand, RANSAC exhibited unsatisfactory results with MSE of 0.0062 and a R^2 of -0.0729. Similarly, SVM, LR, Ridge, and Lasso similarly shown inferior performance, with an MSE of 0.0057 and R^2 values of 0.0039, 0.0036, 0.0036, and 0.0000, respectively. Among the tested models, EN, Lasso, SVM, LR, and Ridge achieved the most favourable outcomes on the test data. Each of these models yielded a MSE of 0.0093 and R^2 values of -0.0014, -0.0000, -0.0096, and -0.0070, respectively. Nevertheless, the DT and ET models exhibited the poorest performance on the test data, as evidenced by their MSEs of 0.0242 and 0.0131, and R^2 values of -1.6146 and -0.4154, respectively.

When applying DT and ET models to the data of AMZN, which operates in the Catalog/Specialty Distribution sector, the training data produced the best results. These models achieved MSE of 0.0000 and a R^2 value of 1.0000. In contrast, EN and RANSAC had the lowest performance on the training data, with MSEs of 20.2486 and 22.2728, and R^2 values of 0.0066 and -0.0927, respectively. However, EN and Lasso produced the most favourable outcomes on the test data, achieving MSEs of 83.1652 and 83.2893, and R^2 values of 0.0070 and 0.0056, respectively. On the other hand, the DT and RT models exhibited the poorest performance on the test data, as evidenced by their MSEs of 106.4541 and 206.5049, and R^2 values of -0.2710 and -1.4656, respectively.

Table II: Performance Summary of Various Regression Models Across Multiple Stocks.

Model Name	PM	MSFT	AAPL	AMSF	ABCB	AHGP	AETI	ACAD	ARRAY	FLWS	AMZN
Support Vector Regression	MSE Train	0.1156	0.0085	0.0316	0.0052	0.0052	0.0143	0.1309	0.0114	0.0057	17.5923
	MSE Test	0.2642	1.1801	0.1708	0.1402	0.1402	0.0077	0.2986	0.0060	0.0093	86.6440
	R^2 Train	0.0158	0.9872	0.6782	0.7915	0.7915	0.0366	0.1148	0.0031	0.0039	0.1369
	R^2 Test	-0.0078	-0.0017	-0.0000	-0.0084	-0.0084	-0.4698	-0.0297	-0.0042	-0.0096	-0.0345
Linear Regression	MSE Train	0.0813	0.0813	0.4339	0.0499	0.0481	0.0487	0.0042	0.0897	0.0057	20.1704
	MSE Test	0.4672	2.5634	0.3906	0.0996	0.0996	0.0567	0.5728	0.0455	0.0093	83.3354
	R^2 Train	0.0047	0.0323	0.0057	0.0059	0.0059	0.0470	0.0307	0.0038	0.0036	0.0105
	R^2 Test	-0.0291	-0.0294	-0.0007	-0.0181	-0.0181	-0.0804	-0.0170	0.0007	-0.0070	-0.0050
Ridge Regression	MSE Train	0.1169	0.6412	0.0977	0.0249	0.0249	0.0142	0.1433	0.0114	0.0057	20.1704
	MSE Test	0.2697	1.2125	0.1709	0.1415	0.1415	0.0057	0.2948	0.0059	0.0093	83.3354
	R^2 Train	0.0047	0.0323	0.0057	0.0059	0.0059	0.0469	0.0307	0.0038	0.0036	0.0105
	R^2 Test	-0.0289	-0.0292	-0.0007	-0.0180	-0.0180	-0.0822	-0.0166	0.0002	-0.0070	0.0050
Lasso Regression	MSE Train	0.1175	0.6624	0.0983	0.0251	0.0251	0.0149	0.1478	0.0114	0.0057	20.2831
	MSE Test	0.2639	1.1805	0.1708	0.1398	0.1398	0.0059	0.2910	0.0059	0.0093	83.2893
	R^2 Train	0.0000	0.0002	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0049
	R^2 Test	-0.0066	-0.0020	-0.0000	-0.0056	-0.0056	-0.1357	-0.0034	-0.0016	-0.0000	0.0056
ElasticNet Regression	MSE Train	0.1174	0.6545	0.0981	0.0250	0.0250	0.0149	0.1462	0.0114	0.0057	20.2486
	MSE Test	0.2675	1.1796	0.1711	0.1410	0.1410	0.0062	0.2904	0.0059	0.0093	83.1652
	R^2 Train	0.0004	0.0121	0.0018	0.0008	0.0008	0.0020	0.0108	0.0002	0.0028	0.0066
	R^2 Test	-0.0203	0.1582	-0.0020	-0.0142	-0.0142	-0.1830	-0.0013	-0.0019	-0.0014	0.0070
Random Forest Regression	MSE Train	0.0233	0.1582	0.0263	0.0055	0.0055	0.0037	0.0260	0.0027	0.0012	3.6827
	MSE Test	0.3238	1.4956	0.1745	0.1846	0.1846	0.0098	0.3368	0.0066	0.0115	206.5049
	R^2 Train	0.8019	0.7612	0.7319	0.7809	0.7809	0.7482	0.8244	0.7637	0.7994	0.8193
	R^2 Test	-0.2351	-0.2695	-0.0217	-0.3279	-0.3279	-0.8821	-0.1612	-0.1108	-0.2429	-1.4656
RANSAC	MSE Train	0.1344	0.6769	0.1133	0.0368	0.0265	0.0179	0.1594	0.0126	0.0062	22.2728
	MSE Test	0.4951	1.3181	0.2032	0.1653	0.1560	0.0060	0.3093	0.0064	0.0107	102.5914
	R^2 Train	-0.1444	-0.0217	-0.1528	-0.4673	-0.0571	-0.2006	-0.0786	-0.1031	-0.0729	-0.0927
	R^2 Test	-0.8887	-0.1188	-0.1901	-0.1890	-0.1221	-0.1464	-0.0666	-0.0806	-0.1545	-0.2249
Decision Tree Regression	MSE Train	0.0000	0.0000	0.0000	0.0000	0.0000	0.0005	0.0000	0.0000	0.0000	0.0000
	MSE Test	0.3236	1.6138	0.1851	0.2146	0.2300	0.0433	0.5316	0.0091	0.0242	106.4541
	R^2 Train	1.0000	1.0000	1.0000	1.0000	1.0000	0.9681	1.0000	1.0000	1.0000	1.0000
	R^2 Test	-0.2344	-0.3698	-0.0838	-0.5436	-0.6543	-7.2923	-0.8331	-0.5301	-1.6146	-0.2710
Extra Tree Regression	MSE Train	0.0000	0.0000	0.0000	0.0000	0.0000	0.0005	0.0000	0.0000	0.0000	0.0000
	MSE Test	0.3250	1.4562	0.1994	0.1923	0.1946	0.0169	0.3585	0.0069	0.0131	104.0198
	R^2 Train	1.0000	1.0000	1.0000	1.0000	1.0000	0.9681	1.0000	1.0000	1.0000	1.0000
	R^2 Test	-0.2399	-0.2361	-0.1676	-0.3826	-0.3994	-2.2333	-0.2363	-0.1688	-0.4154	-0.2420

DT and ET models exhibited the highest level of accuracy on the training data for all organisations, whereas Lasso, LR, and RANSAC displayed the lowest level of performance. In contrast, Lasso, SVR, EN, and RR demonstrated superior performance compared to the other models on the testing data, whereas LR, RANSAC, DT, and ET were shown to be less successful.

The propensity of Decision Tree and Extra Tree models to excessively match the training data elucidates their robust performance during training but limited ability to generalise to testing data. RANSAC, however, exhibited poor performance on both the training and testing data, most likely because it is not well-suited for

time series data. Its random sampling method may not properly capture the underlying trends in this type of data.

Linear Regression, which is based on statistical modelling, did not perform well, especially when there were non-linear correlations between the independent and dependent variables. The Support Vector Regression (SVR) algorithm, known for its resistance to outliers and ability to handle non-linear problems, shown strong and reliable performance, particularly in challenging and intricate situations.

Lasso, utilising L1 regularisation, selectively reduces some coefficients to zero, thereby successfully

conducting feature selection. This is particularly beneficial when dealing with datasets that have a large number of dimensions. Ridge Regression, a statistical technique that shrinks the coefficients without eliminating them entirely, does not actively pick features but effectively mitigates the risk of overfitting. Lasso and Ridge enable the utilisation of intricate models while reducing the risk of overfitting.

ElasticNet is a powerful tool that combines the advantages of Lasso and Ridge. It enhances the stability of Ridge when the data is rotated and provides flexibility in selecting the number of variables. This makes it particularly useful in situations where both feature selection and model stability are important.

Conclusion: In this paper, we conducted a comparative analysis of various machine learning regression models for predicting the next day's opening prices of stocks. Although previous research in this field has primarily focused on predicting the future direction of price movements, our work highlights the significance of precisely anticipating the actual price values. This strategy provides investors with more accurate and detailed information on prospective market behavior, enabling them to make more informed decisions.

The findings of our study indicate that all the assessed models, including SVR, Lasso, ElasticNet, Ridge Regression, Decision Tree, Extra Tree, and RANSAC, possess the ability to forecast stock prices to varying degrees. Our analysis revealed that SVR, Lasso, ElasticNet, and Ridge Regression demonstrate superior performance compared to other models in terms of MSE and R^2 scores. The findings emphasize the efficacy of utilizing historical stock prices to forecast future values, hence strengthening the potential of computational intelligence techniques in financial prediction. Our research further emphasises the crucial importance of accurate price prediction in delivering important indications to investors, hence improving their strategic decision-making in the stock market.

REFERENCES

- [1] Qian, B., Rasheed, K. (2007). Stock market prediction with multiple classifiers. *Applied Intelligence*, 26(1), 25-33.
- [2] Shah, D., Isah, H., Zulkernine, F. (2019). Stock market analysis: A review and taxonomy of prediction techniques. *International Journal of Financial Studies*, 7(2), 26.
- [3] Pathak, A., Shetty, N. P. (2019). Indian stock market prediction using machine learning and sentiment analysis. In *Computational Intelligence in Data Mining* (pp. 595-603). Springer, Singapore.
- [4] Patel, J., Shah, S., Thakkar, P., Kotecha, K. (2015). Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. *Expert systems with applications*, 42(1), 259-268.
- [5] Kim, K. J., Lee, W. B. (2004). Stock market prediction using artificial neural networks with optimal feature transformation. *Neural computing applications*, 13(3), 255-260.
- [6] Ticknor, J. L. (2013). A Bayesian regularized artificial neural network for stock market forecasting. *Expert systems with applications*, 40(14), 5501-5506. [7] Chung, H., Shin, K. S. (2018). Genetic algorithm-optimized long short-term memory network for stock market prediction. *Sustainability*, 10(10), 3765. [8] Wen, M., Li, P., Zhang, L., Chen, Y. (2019). Stock market trend prediction using high-order information of time series. *Ieee Access*, 7, 28299-28308.
- [9] Cao, J., Li, Z., Li, J. (2019). Financial time series forecasting model based on CEEMDAN and LSTM. *Physica A: Statistical mechanics and its applications*, 519, 127-139.
- [10] Fama, E. F. (1970). Efficient capital markets: A review of theory and empirical work. *The journal of Finance*, 25(2), 383-417.
- [11] Al-Hmouz, R., Pedrycz, W., Balamash, A. (2015). Description and prediction of time series: A general framework of granular computing. *Expert Systems with Applications*, 42(10), 4830-4839.
- [12] Booth, A., Gerding, E., McGroarty, F. (2014). Automated trading with performance weighted random forests and seasonality. *Expert Systems with Applications*, 41, 3651-3661.
- [13] Sheta, A. F., Ahmed, S. E. M., Faris, H. (2015). A comparison between regression, artificial neural networks and support vector machines for predicting stock market index. *Soft Computing*, 7(8), 2.
- [14] Roy, S. S., Mittal, D., Basu, A., Abraham, A. (2015). Stock market forecasting using LASSO linear regression model. In *Afro-European Conference for Industrial Advancement* (pp. 371-381). Springer, Cham.
- [15] Aseervatham, S., Antoniadis, A., Gaussier, E., Burlet, M., Denneulin, Y. (2011). A sparse version of the ridge logistic regression for large-scale text categorization. *Pattern Recognition Letters*, 32(2), 101-106.
- [16] Nair, B. B., Minuvarthini, M., Sujithra, B., Mohandas, V. P. (2010, October). Stock market prediction using a hybrid neuro-fuzzy system. In *2010 International Conference on Advances in*

- Recent Technologies in Communication and Computing (pp. 243-247). IEEE.
- [17] Gunter, S., Senyurt, A. (2012). Stock market movement direction prediction using tree algorithms. In 3rd International Symposium on Sustainable Development (pp. 374-379).
- [18] Thakur, S., Vadpey, T., Ayyar, S. Predicting Stock Prices and Analyst Recommendations.
- [19] Nunno, L. (2014). Stock market price prediction using linear and polynomial regression models. Computer Science Department, University of New Mexico: Albuquerque, NM, USA.
- [20] Wu, M-C., Lin, S-Y., Lin, C-H.,(2006) , “An effective application of decision tree to stock trading”, *Expert Systems with Applications*, 31, pp.270–274.
- [21] Huang, K.Y, and Jane, C.-J., (2009), “A hybrid model for stock market forecasting and portfolio selection based on ARX, grey system and RS theories”, *Expert Systems with Applications*, 36, pp.5387–5392
- [22] Yang, Y., Yang, H. (2014). Adaptive elastic net and Separate Selection from Least Squares for ultra-high dimensional regression models. *arXiv preprint arXiv:1410.3566*.
- [23] Gharehchopogh, F. S., Bonab, T. H., Khaze, S. R. (2013). A linear regression approach to prediction of stock market trading volume: a case study. *International Journal of Managing Value and Supply Chains*, 4(3), 25.
- [24] Sinha, U., Vidyala, N., Saxena, S., Sathyaraj, R. (2017). Comparative Study of Stock Prediction System using Regression Techniques. *International Journal of Advanced Research in Computer Science*, 8(3).
- [25] Kazem, A., Sharifi, E., Hussain, F. K., Saberi, M., Hussain, O. K. (2013). Support vector regression with chaos-based firefly algorithm for stock market price forecasting. *Applied soft computing*, 13(2), 947-958.
- [26] Guresen, E., Kayakutlu, G., Daim, T. U. (2011). Using artificial neural network models in stock market index prediction. *Expert Systems with Applications*, 38(8), 10389-10397.
- [27] Weng, B., Ahmed, M. A., Megahed, F. M. (2017). Stock market one-day ahead movement prediction using disparate data sources. *Expert Systems with Applications*, 79, 153-163.
- [28] Di, X. (2014). Stock trend prediction with technical indicators using SVM. Stanford: Leland Stanford Junior University.
- [29] Zhang, K., Zhong, G., Dong, J., Wang, S., Wang, Y. (2019). Stock market prediction based on generative adversarial network. *Procedia computer science*, 147, 400-406.
- [30] Yang, H., Chan, L., King, I. (2002, August). Support vector machine regression for volatile stock market prediction. In *International conference on intelligent data engineering and automated learning* (pp. 391-396). Springer, Berlin, Heidelberg.
- [31] Gururaj, V., Shriya, V. R., Ashwini, K. (2019). Stock market prediction using linear regression and support vector machines. *Int J Appl Eng Res*, 14(8), 1931-1934.
- [32] Al-Jawarneh, A. S., Ismail, M. (2021). Elastic-Net Regression based on Empirical Mode Decomposition for Multivariate Predictors. *Pertanika Journal of Science Technology*, 29(1).
- [33] Deng, S., Wang, C., Wang, M., Sun, Z. (2019). A gradient boosting decision tree approach for insider trading identification: An empirical model evaluation of China stock market. *Applied Soft Computing*, 83, 105652.
- [34] Tang, X. B., Liu, G. C., Yang, J., Wei, W. (2018). Knowledge-based financial statement fraud detection system: based on an ontology and a decision tree. *KO KNOWLEDGE ORGANIZATION*, 45(3), 205-219.
- [35] Yenikara, A., Balib, M., Babuc, N. (2019). EMP-SA: Ensemble Model based Market Prediction using Sentiment Analysis.
- [36] Drucker, H., Burges, C. J., Kaufman, L., Smola, A., Vapnik, V. (1996). Support vector regression machines. *Advances in neural information processing systems*, 9.
- [37] Panwar, B., Dhuriya, G., Johri, P., Yadav, S. S., Gaur, N. (2021, March). Stock market prediction using linear regression and SVM. In *2021 International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)* (pp. 629-631). IEEE.
- [38] McDonald, G. C. (2009). Ridge regression. *Wiley Interdisciplinary Reviews: Computational Statistics*, 1(1), 93-100.
- [39] Roth, V. (2004). The generalized LASSO. *IEEE transactions on neural networks*, 15(1), 16-28.
- [40] Tay, J. K., Narasimhan, B., Hastie, T. (2023). Elastic net regularization paths for all generalized linear models. *Journal of statistical software*, 106.
- [41] Suthaharan, S., Suthaharan, S. (2016). Decision tree learning. *Machine Learning Models and Algorithms for Big Data Classification: Thinking with Examples for Effective Learning*, 237-269.
- [42] Borup, D., Christensen, B. J., Muhlbach, N. S., Nielsen, M. S. (2023). Targeting predictors in random forest regression. *International Journal of Forecasting*, 39(2), 841-868.

- [43] Alkawaz, A. N., Abdellatif, A., Kanesan, J., Khairuddin, A. S. M., Gheni, H. M. (2022). Day-ahead electricity price forecasting based on hybrid regression model. *IEEE Access*, 10, 108021-108033.
- [44] Yan, F., Guo, Z., Yu, Y., Zhang, C., Liu, X. (2023). Vegetable Sales and Pricing Correlations: Insights from a RANSAC Regression Analysis. *life*, 7(9),1-7.
- [45] Smith, H. (2024). The utilization of Artificial Intelligence to predict Apple Inc. stock price. *Journal of High School Science*, 8(3), 16-29.
- [46] S,ekeroglu, G., Karabo~ ga, E. N. C., Kızılo~ glu, E.,~ Karaboga, K. (2024). Which Stock Market is More Predictable? Comparison of Machine Learning~ and Deep Learning Algorithms for Stock Price Prediction of Company with Dual-Listed Stocks. *ICAM'24*, 61.
- [47] Easton, P. D., Kapons, M. M., Monahan, S. J., Schutt, H. H., Weisbrod, E. H. (2024). Forecasting earnings using K-nearest neighbors. *The Accounting~ Review*, 99(3), 115-140.
- [48] Bloch, G. (2024). Data Analysis for Stock Price Prediction